



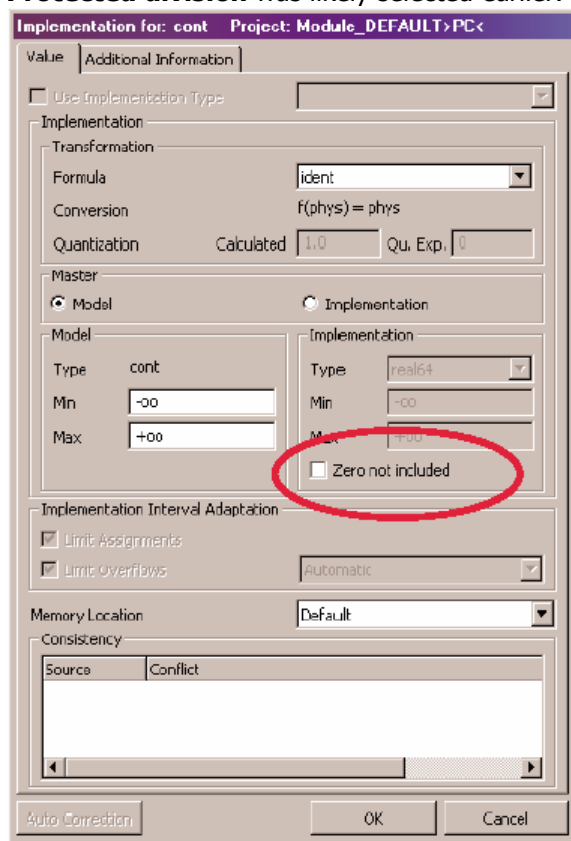
## Check Box "Zero Not Included" in the Implementation-Editor

### PROBLEM DESCRIPTION

The code generator is supposed to optimize **Protected division** from within the code in order to avoid producing superfluous code.

### SOLUTION

If the check box **Zero not included** is checked, the code generator assumes that value "0" does not occur during runtime. In this case, no code is generated for **Protected division** even though the option **Protected division** was likely selected earlier.

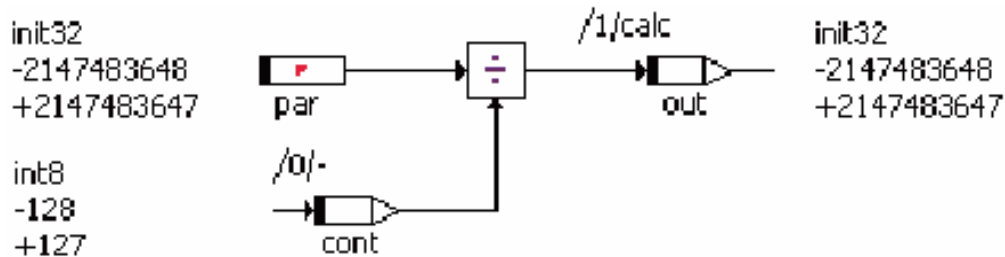


The user assumes the responsibility to exclude "0" if she/he selects the check box **Zero not included**.

If a value of "0" does occur on occasion, a runtime error might occur as the following example demonstrates.

Example for the interaction between **Zero not included** and **Protected division** (par = 100; cont = 0):





•Variant 1 (Cont: "Zero not included" – OFF, "Protected division" – OFF):

Generated C code: `out = par / cont;`

Experiment:



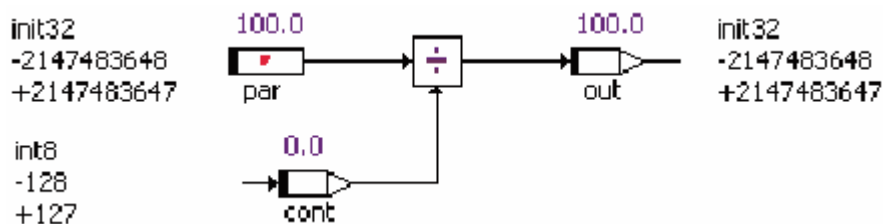
In this case, a division by zero may occur.

•Variant 2 (Cont: "Zero not included" – OFF, "Protected division" – ON)

Generated C code:

```
out = (cont == (sint8)0) ? (asdWriteUserError ("Run Time Error: division by zero with variable <cont> in component <XY>\n"), par) : par / cont;
```

Experiment:



•Variant 3 (Cont: "Zero not included" – ON, "Protected division" – OFF)

Generated C code: `out = par / cont;`

Experiment: error message Variant 1

•Variant 4 (Cont: "Zero not included" – ON, "Protected division" – ON)





Generated C code: `out = par / cont;`

Experiment: error message Variant 1

**APPLIES TO PRODUCTS**

ASCET-MD

**ADDITIONAL KEY WORDS**

arithmetic, code efficiency

